A comparison of artificial neural networks used for river forecasting

C. W. Dawson, R. L. Wilby

Hydrology and Earth System Sciences, 3(4), *529–540* (1999) © EGS

Hydrology & Earth System Sciences

A comparison of artificial neural networks used for river flow forecasting

Christian W. Dawson[1] and Robert L. Wilby[2]

[1] Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, UK
[2] National Center for Atmospheric Research, Boulder, Colorado 80307-3000, USA
[2] School of Environmental and Applied Science, University of Derby, Kedleston Road, Derby, DE22 1GB, UK
email address for corresponding author: C.W.Dawson1@lboro.ac.uk

Abstract

This paper compares the performance of two artificial neural network (ANN) models—the multi layer perceptron (MLP) and the radial basis function network (RBF)—with a stepwise multiple linear regression model (SWMLR) and zero order forecasts (ZOF) of river flow. All models were trained using 15 minute rainfall-runoff data for the River Mole, a flood-prone tributary of the River Thames, UK. The models were then used to forecast river flows with a 6 hour lead time and 15 minute resolution, given only antecedent rainfall and discharge measurements. Two seasons (winter and spring) were selected for model testing using a cross-validation technique and a range of diagnostic statistics. Overall, the MLP was more skillful than the RBF, SWMLR and ZOF models. However, the RBF flow forecasts were only marginally better than those of the simpler SWMLR and ZOF models. The results compare favourably with a review of previous studies and further endorse claims that ANNs are well suited to rainfall-runoff modelling and (potentially) real-time flood forecasting.

Introduction

According to Hsu *et al.* (1995), artificial neural network (ANN) approaches to rainfall-runoff modelling are more efficient than conventional flow forecasting models whenever explicit knowledge of the hydrological balance is not required and when the system may be treated as a black-box. However, many ANN configurations are available and this paper compares two of the most favoured architectures: the multi layer perceptron (MLP) and the radial basis function network (RBF) applied to river flow forecasting. While the MLP has been evaluated for rainfall-runoff modelling in a number of studies (e.g. Braddock *et al.*, 1998; Campolo *et al.*, 1999; Zealand *et al.*, 1999), the RBF has been used in relatively few cases (e.g. Jayawardena *et al.*, 1998; Mason *et al.*, 1996).

For a flood forecasting system (FFS) to be effective, it must provide flood warnings with a reasonable lead time and at an appropriate temporal granularity. Previous studies have noted that the accuracy of ANN model forecasts decrease as the lead times increase. For example, Campolo *et al.*, (1999) forecasted flows in the River Tagliamento (catchment area 2480 km^2) up to 10 hours ahead at one hour intervals but reported deteriorating results after 5 hours. In comparison, this paper describes the development of ANNs for forecasting runoff in the River Mole (catchment area 142 km^2) with a forecast lead time of 6 hours and 15 minute temporal granularity. In all catchments, the forecasting skill is ultimately constrained by the choice of ANN inputs, as well as by factors such as dominant storm characteristics and time lag between rainfall-runoff. This study follows closely the methodology of Dawson and Wilby (1998) and takes that work further by evaluating radial basis function networks when applied to the same catchment.

To establish the true merit of ANNs relative to simpler statistical techniques, comparisons are also made between the forecasting skill of the two ANNs and those of a stepwise multiple linear regression (SWMLR) model and Zero Order Forecasts (ZOFs). Furthermore, in an earlier study Dawson and Wilby (1998) identified problems transferring ANNs to runoff seasons outside of the training set. This problem is overcome in this study by stratifying the data by season and then applying a cross validation technique. Finally, suggestions are outlined for future research in ANN rainfall-runoff modelling and real-time flood forecasting.

Artificial neural networks

Research in artificial neural networking has experienced a renaissance since the rediscovery and popularisation of the backpropagation algorithm by Rumelhart and McClelland (1986). Prior to this publication, it was very difficult to train neural networks of the size needed for most practical applications. This important breakthrough subsequently stimulated research in two main areas: the development and advancement of artificial neural networks *per se* (i.e. architectures, training algorithms, etc.), and the application of these tools to a variety of complex management problems. However, in any given situation, the effective application of ANNs still requires an appreciation of the relative merits of different networks, as well as how best to train them. For example, having chosen a suitable neural network architecture, one still needs to determine the appropriate network 'size' and the most efficient training algorithm (e.g. Dai and MacBeth, 1997). Furthermore, one must also pre- and post-process the data (for cross validation and normalisation/standardisation) and select suitable training periods. While some of these factors are determined dynamically—using appropriate modifications to training algorithms—many decisions must still be made through a process of trial and error. A full discussion of these topics is beyond the scope of the present paper and interested readers are directed towards texts such as Bishop (1995) and Gallant (1993).

NETWORK STRUCTURE

Two of the most common neural network structures were chosen for modelling: the MLP and the RBF network. These are both feed forward networks, typically consisting of three connected layers of *neurons* (as shown in Fig. 1). The number of neurons in the input and output layer is specified by the problem to which the network is applied (i.e. the number of predictors and predictands respectively). In this study, seven nodes were used in the input layer and one node, representing flow, was used in the output layer. The neural network engineer must specify the number of neurons in the hidden layer (and, in some cases, the number of hidden layers). If there are insufficient neurons in the hidden layer, the network may be unable to model the underlying function because it has insufficient parameters to map all points in the training data. Conversely, if there are too many neurons, the network has, in effect, too many free parameters and may overfit the data and hence lose its ability to generalise. While it is possible to determine an 'optimum' number of neurons to use in the hidden layer, using pruning or constructive algorithms such as cascade correlation (e.g. Hirose *et al.*, 1991), these algorithms can retard training significantly by introducing additional computations. Shamseldin (1997) claims that the best way to determine an appropriate number of neurons in the hidden layer is via trial and error. Accordingly, networks with 5, 10, 20 and 30 hidden neurons are considered here.

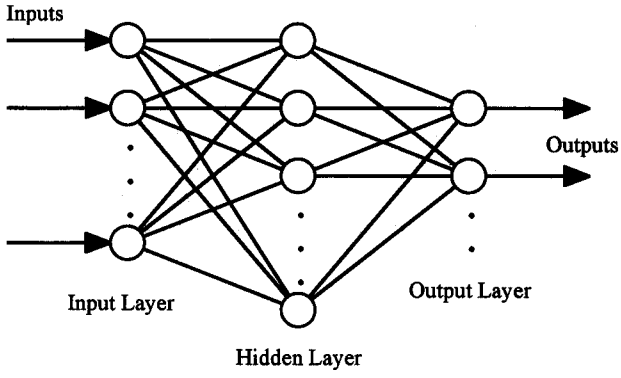

Fig. 1. *A feed forward artificial neural artwork structure*

ACTIVATION FUNCTIONS

Inputs to the network (predictors) are passed from the input layer of neurons, through the hidden layer of neurons, to the output layer (see Fig. 1). Neurons in the input layer do no more than disperse all inputs to each neuron in the hidden layer. The network then operates by applying weights to values as they pass from one layer to the next. Each neuron in the hidden and output layers computes an output, based on the weighted sum of all its inputs, according to an activation function. These functions may be logistic sigmoid, linear, threshold, Gaussian or hyperbolic tangent functions depending on the type of network and training algorithm employed.

The network is trained by adjusting the weights that link individual neurons. This is accomplished by presenting the network with a number of training samples (a calibration data set)—each one of which consists of a specific input pattern and corresponding 'correct' output response. Depending on the nature of the training algorithm used, it may be necessary to present a network with the calibration data repeatedly (a number of epochs) until it 'learns' the underlying function being modelled. However, care must be taken to ensure that the network does not become overly specified to the calibration data set and thus lose its ability to generalise to problems it has not encountered before. Various techniques may be employed to avoid over training; they include regularisation theory which attempts to smooth network mappings (Bishop, 1995), and cross validation using an independent test set (Braddock *et al.*, 1998).

THE MULTI LAYER PERCEPTRON

The MLP is the most popular neural network architecture in use today. It assumes that the unknown (rainfall-runoff) function is represented by a multi-layer, feed forward net-

work of sigmoid units (shown in Fig. 1). The logistic sigmoid activation function (Eqn. 1; where x represents the weighted sum of inputs to the neuron and $f(x)$ the neuron's output) is used because it is continuous, and it is easy to compute, as is its derivative. In this study, neurons in both the hidden and output layers were represented by this function.

$$f(x) = \frac{1}{1+e^{-x}} \quad (1)$$

In this study, the MLP was trained using the error backpropagation algorithm. This popular algorithm works by iteratively changing a network's interconnecting weights such that the overall error (i.e. between observed values and modelled network outputs) is reduced. This is achieved by searching the network's 'weight space' or error function. The error function is an error surface in n-dimensional space corresponding to a mapping of the network's weight vector to the network's overall error. The purpose of training is to search this error surface, by adjusting a network's weights, such that an acceptable error minimum is found. Training is initiated from randomly determined regions of the error surface. The algorithm then proceeds by directing weight changes down error gradients based on the first order derivative of the error function. Step changes are made to weights as each training example is presented to the network (an epoch). The 'training rate' parameter affects the size of step taken through weight space at each training iteration. If the rate is too large, training can oscillate from one non-optimal set of weights to another; if the rate is too small training may be trapped in a local error minimum or sub-optimal solution.

There are many other training techniques available (e.g. line search algorithms, such as conjugate gradients, and Newton's Method [Bishop, 1995; Battiti, 1992]). For example, Thirumalaiah and Deo (1998) compared the results of an MLP rainfall-runoff model trained using three different training algorithms (error backpropagation, conjugate gradients and cascade correlation). While cascade correlation provided rather poor results, little difference was found in network performance between the backpropagation and conjugate gradient training approaches.

For the present purpose, the error backpropagation algorithm was adapted in two ways. Firstly, momentum (which keeps weight changes on a faster, more even path and helps to avoid local minima) was used in an attempt to speed convergence to an error minimum (Gallant, 1993). Momentum is controlled using a 'momentum rate' which must be less than unity for convergence. Secondly, the training rate was adjusted dynamically to prevent the optimisation becoming caught in a local error minimum (Dawson, 1996; Magoulas *et al.*, 1997). The training rate was varied between 0.1 to 0.01 and the momentum rate was fixed at 0.9. A trial and error approach was used to determine the most effective training duration (or number of epochs) for the MLP by comparing its performance with a validation data set. Evaluations were made of MLPs trained for 500, 1000, 2000, 3000, 4000 and 5000 epochs.

RADIAL BASIS FUNCTION

While the structure of the RBF is identical to the MLP, the RBF simulates the unknown rainfall-runoff function using a network of Gaussian basis functions in the hidden layer (Eqn. 2) and linear activation functions in the output layer. In Eqn. 2, x represents the weighted sum of inputs to the neuron, σ is the sphere of influence or width of the basis function, and $f(x)$ is the corresponding output from the neuron.

$$f(x) = e^{-x^2/2\sigma^2} \quad (2)$$

Training an RBF involves two stages. Firstly, the basis functions are established using an algorithm to cluster data in the training set. In this case, a k-means clustering algorithm was used. K-means clustering involves sorting all objects into a defined number of groups by minimising the total squared Euclidean distance for each object with respect to its nearest cluster centre. However, other techniques, such as Kohohen self organising maps, orthogonal least squares and MaxiMin algorithms might also be used (Song, 1996).

Secondly, the weights linking the hidden layer to the output layer were calculated directly using simple matrix inversion (singular value decomposition) and matrix multiplication. Because of the direct calculation of the weights in the RBF, it is far quicker to train than an equivalent MLP (seconds rather than hours using a 233MHz processor).

As a starting point, RBFs were developed with the same number of basis functions as hidden nodes in the MLPs to enable a direct comparison to be made between the two network types (i.e. 5, 10, 20 and 30 basis functions). Because the RBF can be trained very quickly, this experiment was extended to 40 basis functions to assess if this would lead to an improvement in forecasting accuracy. However, as results showed, this did not lead to any improvement.

Methodology

The methodology initially employed here follows that of Dawson and Wilby (1998). The Environment Agency (Thames region) provided 15-minute rainfall-runoff data for the River Mole for all of 1994. These data include rainfall (mm), recorded at 15 minute intervals at the Burstow raingauge, and 15 minute flow measurements made at Kinnersley Manor (in cumecs). The River Mole, a flood-prone catchment of approximately 142 km^2, is a lowland tributary of the River Thames which drains largely

impervious soils in the vicinity of Gatwick airport. Note that the relatively small basin area and 'flashy' response of the flow to heavy rainfall present significant forecasting challenges.

DATA PRE-PROCESSING

Before ANNs are trained, input data must be pre-processed. It was intended to develop models to predict river flow at a time t_0 using observations that had occurred previously at times t_{-n} (representing a time n minutes before t_0). Exploratory analysis of the rainfall-runoff data were undertaken using two methods: windowing and auto-regression.

Windowing involves the use of antecedent flows and rainfall at times t_{-1}, t_{-2}, . . . t_{-n} as direct inputs into the model (as in Hall and Minns [1993], and Abrahart and Kneal [1997]). However, initial experimentation with this approach, with a 6 hour lead time, produced very poor results because current runoff is related to the cumulative affect of previous flow and rainfall (surrogate measures of the state of the basin), rather than discrete 15 minute events occurring 6 hours earlier. Although windowing can produce acceptable results with either shorter lead times or larger temporal granularity, individual 15 minute flow and rainfall events were found to have limited predictive power for the River Mole at the specified 6 hour lead. As a consequence, the moving window approach was disregarded and an alternative technique after that of Refenes *et al.* (1997) was adopted. This technique involves the use of ARMA (Auto Regressive Moving Average) models to select input variables and lag times, and a stepwise regression method for determining appropriate combinations of inputs from these variables. Again, an arbitrary lead time of 6 hours (t_{-360}) was chosen as the initial point for the analyses.

Table 1 shows the lagged variables identified by the ARMA models (with their corresponding correlation coefficients, Pearson r) as significant ($p<0.005$) predictors of flow at time t_0. These were selected by varying the lag interval and moving average periods within the ranges 6-12 hours and 2-96 hours respectively. Individual precipitation events exceeding specified duration and intensity thresholds are also known to affect flow in the River Mole significantly (Dawson and Wilby, 1998). Storm events were defined *a priori* as periods of rainfall lasting over one hour or having intensities exceeding 0.5 mm in any 15 minute interval. Hydrologically, this variable represents the 'filtering' of small (<1 mm) and/or isolated precipitation events by canopy interception, surface depression storage and subsequent evaporation. In the case of the Mole, the optimal lag-interval for the storm parameter was found to be 27 hours, compared with 15 hours for the instantaneous rainfall totals, and 6 hours for the moving average rainfall. However, the latter was by far the most strongly correlated with discharge, implying that the chosen forecast lead time of 6 hours is of the same temporal order as the catchment response time. Thus, the seven factors, shown in Table 1, represent the most significant predictors of flow at time t_0 that could be identified from the rainfall-runoff records, and constitute the inputs used to train and evaluate the two ANNs.

Table 1. Predictors used to hindcast River Mole discharges at t_0

Predictors	Lag (minutes)	Correlation Coefficient
15 minute instantaneous discharge	360	0.891
30 minute moving average of discharge	360	0.889
Previous 24 hours total discharge	360	0.468
15 minute instantaneous rainfall total	900	0.417
20 hour moving average of rainfall	360	0.841
Previous 24 hours total rainfall	360	0.727
Storm events	1620	0.551

DATA NORMALISATION/STANDARDISATION

The gradient descent algorithm (error backpropagation) used to train the MLP is particularly sensitive to the scale of data used. Due to the nature of this algorithm, large values slow training because the gradient of the sigmoid function at extreme values approximates zero. Also, because of the nature of the logistic activation function used in the output layer, outputs from the network are constrained to the range [0,1]. In addition, because each predictor can cover a different range of values, it is prudent to rescale each input to a common range so that one predictor does not dominate all others.

To avoid these problems, input and output data in both the training and test sets were rescaled using an appropriate transformation. In general, data may be rescaled to the interval [–1,1], [–0.9,0.9], [0.1,0.9], or [0,1] (referred to as *standardization*) depending on the network activation functions employed. Another approach is to rescale values to a Gaussian function with a mean of 0 and unit standard deviation (referred to as *normalization*). The advantage of using [0.1, 0.9] for runoff modelling is that extreme (high and low) flow events occurring outside the range of the calibration data may be accommodated. Hsu *et al.* (1995) used this approach to avoid the problem of output signal saturation (i.e. truncation) that can be encountered in ANN applications. An alternative approach involves the use of changes in flow rather than absolute flows to avoid the problem of saturation, but Minns and Hall (1997) reported only limited gains from this refinement.

Both data rescaling procedures have been considered; standardizing data to the range [0.1,0.9] provided the most accurate results. For the RBF, it is particularly important to standardize all input variables so that they span similar

ranges (Bishop, 1995). Hence, data used in training and testing the RBF were also standardized to [0.1,0.9].

CROSS VALIDATION

Because 15 minute rainfall-runoff data for an entire year represent a large data set (35040 values) and embrace seasonal variations in catchment properties, care must be taken when assigning the data to calibration and validation sub-sets. For example, Dawson and Wilby (1998), using data drawn from different seasons, concluded that in the absence of parameters describing seasonal variations in catchment rainfall-runoff responses (due to, for example, soil moisture or infiltration rate variations), the ANN should ideally be validated against data from the same season as the calibration period.

Therefore, the data were split into two sets—the first 100 days of 1994 (representing winter and early spring, 9495 data points) and the last 100 days of 1994 (representing autumn and early winter, 9495 data points). These seasonal data sets were then used to validate the two ANN techniques using a cross validation approach (Bishop, 1995). Cross validation involves splitting the available data into S equal sized segments. Because of strong autocorrelations between successive data points, these segments were created at random rather than being drawn sequentially. This went some way to ensuring that the validation and calibration data sets did not contain near identical data points which would have led to a less rigorous test of the models. The networks were then trained using all data in S-1 of these segments and tested on the remaining segment of independent data. This procedure is repeated S times so that S networks are trained and tested for each network type and configuration. This ensures that each data segment is used only once for validation. Thus, when the validation segments are recombined, one has a validation set that is equal to the entire data set.

Cross validation enables one to identify, more objectively, the configuration of the most accurate seasonal model. One would then use this information to retrain the chosen ANN on all the available data before it is implemented in a practical situation.

In this study, the two seasonal data sets were each split randomly into 5 segments, a typical value for S (see Schalkoff, 1997). The networks were then trained on 4 segments (7596 data points) and tested on the remaining segment (1899 data points). When the training and test cycle was completed for the 5 segments, the ANN model had produced a set of 9495 validation points.

COMPARATIVE MODELS

In addition to the neural network models, it was felt that comparisons should be made with less computationally demanding models. To this end, stepwise multiple linear regression models (SWMLR) were produced for both seasons based on the cross validation data sets (Eqn. 3; in which predictand y is based on predictors $x_{1...n}$). The stepwise procedure adds or removes predictors to a model during a series of iterations. Predictors are added or removed based on the partial F-ratio at each stage of the procedure until no additional benefit is gained from adding or removing variables. Readers are directed towards texts such as Mendenhall and Sincich (1995) for a more thorough discussion of this technique.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \quad (3)$$

Because of the way the predictors were identified for the two seasonal data sets, the stepwise procedure did not disregard any of the input parameters. This was because the set of predictors had been optimally selected beforehand. However, as presented in the following section, the SWMLR models still produced rather disappointing results, due to their underlying assumption of linearity which does not necessarily exist between each variable.

Additionally, the performance of the ANNs was compared with Zero Order Forecasts (ZOFs) in which the measured flow at t_{-360} was taken as the best estimate of flow at t_0. Both alternatives were evaluated in the same way as the two neural network models and comparable diagnostics are presented below.

PERFORMANCE MEASURES

Survey of recent literature describing ANN applications to rainfall-runoff modelling (Table 4) suggests a general lack of a modelling protocol. Firstly, there is no convention for the error measures that are employed (e.g. mean square errors, relative errors and so on). Secondly, the wide variety of catchments studied (in terms of area, topography, land-use, climate regime, etc.) precludes direct comparison of ANN performance in each case. Thirdly, several different measures of flow are employed, such as discharge (Dawson and Wilby, 1998), water level (Campolo *et al.*, 1999; Jayawardena *et al.*, 1997), and rates of change of discharge (Minns and Hall, 1997). Finally, there are differences between the studies in terms of the lead time and temporal granularity of the flow forecasts.

Four of the most commonly employed error measures were calculated: the root mean square error (RMSE); the mean square relative error (MSRE); the coefficient of efficiency (CE); and the coefficient of determination (r^2). According to Karunanithi *et al.* (1994), square errors provide a reliable measure of goodness of fit at the high flows, whilst relative errors are biased towards moderate flows. CE and r^2 also provide useful comparisons with other studies since they are independent of the scale of data used (i.e. flow, catchment, temporal granularity, etc.). However, CE is referred to by some authors as R^2 and care must be taken not to confuse this with the coefficient of determination, r^2. The CE is considered a better measure of the

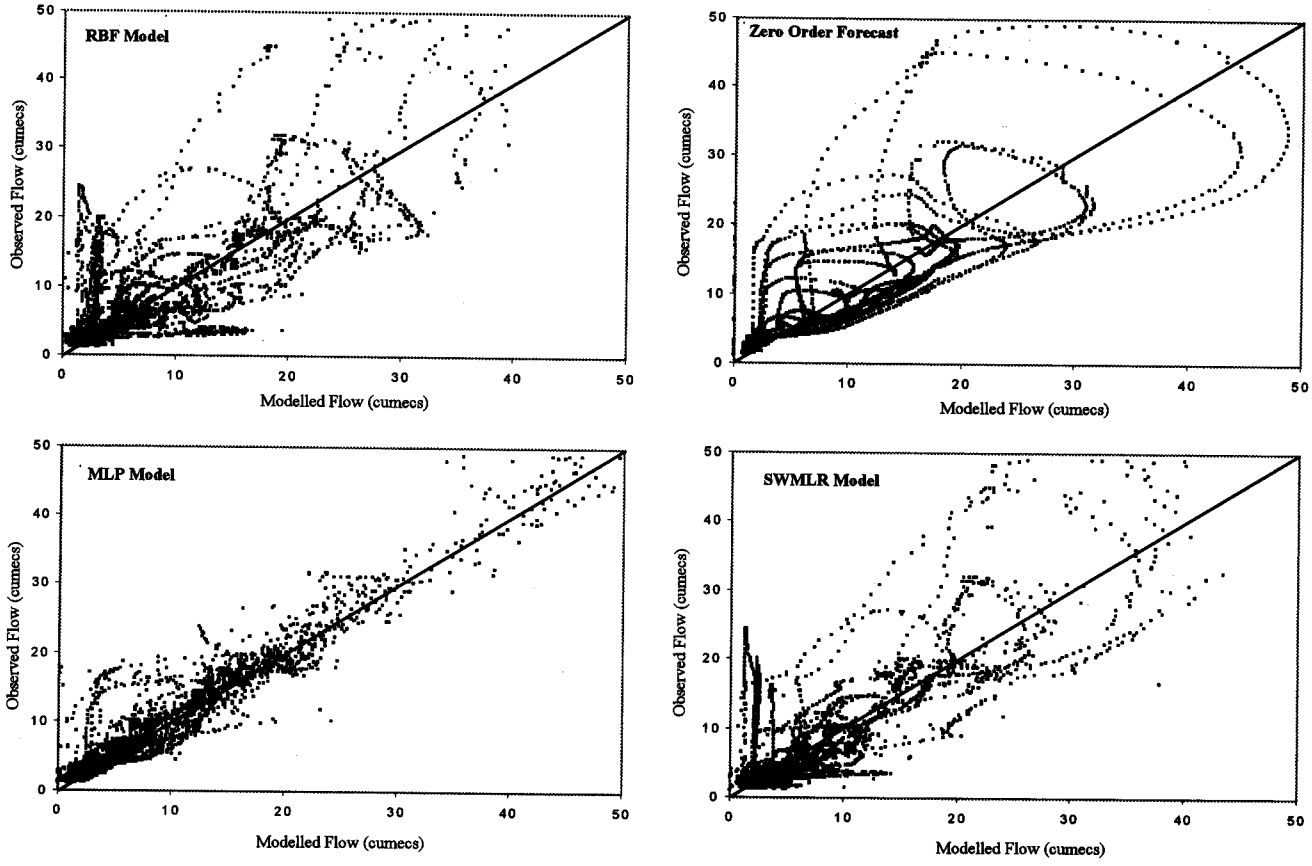

Fig. 2. *Validation results for winter/early spring*

goodness-of-fit than the coefficient of determination because the former is sensitive to differences in the observed and modelled means and variances, whereas the latter is insensitive to additive and proportional differences between the model simulations and observations (Legates and McCabe, 1999).

Results

In all cases, the optimal number of hidden neurons in the MLP was found to be 20. Coincidentally, 20 basis functions in the RBFs were also found to produce the most accurate models. Experimentation, which involved adding more basis functions to the RBFs, produced worse results due to overfitting the calibration data. In the case of the MLP, the best results were those obtained using a training period of 2000 epochs, noting that the RBF was not trained iteratively and so effectively had a single epoch.

WINTER/EARLY SPRING

Table 2 provides a summary of the results of the four models using the cross validation technique for winter/early spring. In all cases, the MLP was more accurate than the RBF, ZOF and SWMLR models. According to Shamseldin's (1997) scale, the MLP was 'very satisfactory', the RBF and SWMLR models were 'fairly good', and the ZOF was 'unsatisfactory'. While the RBF appears more accurate than the SWMLR model (using the MSRE criterion), according to the other error measures it is only marginally better.

Figure 2 shows a scatterplot of the forecast versus observed flows for each model. These plots give a clear indication of the relative skill of each model across the full range of flows, and highlight the superiority of the MLP. However, for flood forecasting purposes, the skill of each model at predicting the largest instantaneous flows is clearly of greatest concern. Accordingly, Fig. 3 shows the time series of discharge forecast by all models during the first two weeks of 1994. Once again the MLP forecasts the timing and magnitude of flood events very closely and note

Table 2. Comparative results for late winter / spring

Model	RMSE (m^3/s)	MSRE	CE	r^2
MLP	1.617	0.059	93%	0.928
RBF	2.712	0.079	81%	0.804
SWMLR	2.758	0.138	80%	0.798
ZOF	3.289	0.060	72%	0.730

that the three anomalous 'spikes' (5 January, 11 January and 12 January) were caused by missing data. The RBF and SWLMR models, on the other hand, while following the general pattern of flow, underestimated the magnitude of peak flows, and in some cases forecast peaks up to 8 hours late. In the case of the RBF model, this was attributed to inefficiencies in the choice of cluster centres by the k-means clustering algorithm.

Table 3 Comparative results for late autumn / winter

Model	RMSE (m^3/s)	MSRE	CE	r^2
MLP	1.147	0.080	97%	0.964
RBF	2.396	0.160	86%	0.848
SWMLR	2.396	0.235	86%	0.859
ZOF	2.942	0.076	79%	0.794

AUTUMN/EARLY WINTER

Table 3 and Fig. 4 show the validation performance of the models using the autumn/early winter data set. Once again, for all models the MLP produced better forecasts of flow than the RBF, SWMLR and ZOF models. The CE in the first case was 97% and the MSRE was 0.080. As before, the RBF provides only marginally better results than the multiple regression model and worse results at moderate flows than the ZOF (as indicated by the MSRE statistic).

Figure 5 focuses on a single flood event which occurred during the first two weeks of December 1994 (the largest on record). As before, the MLP provides a convincing forecast of the flood peak while the RBF and SWMLR models predict the peak around 8 hours too late. This suggests that the significance assigned to previous flows by RBF and SWMLR was too low since the response time of the river depends on the state of saturation of the basin prior to (and/or during the evolution of) the flood event – information that must be supplied by antecedent flows.

CONFIDENCE LIMITS AND UNCERTAINTY

When producing flow forecasts it is informative to provide accompanying measures of confidence based on the model being used. Using the standard error of the least squares regression line between modelled flow and observed flow (i.e. the regression of observed on modelled flow),

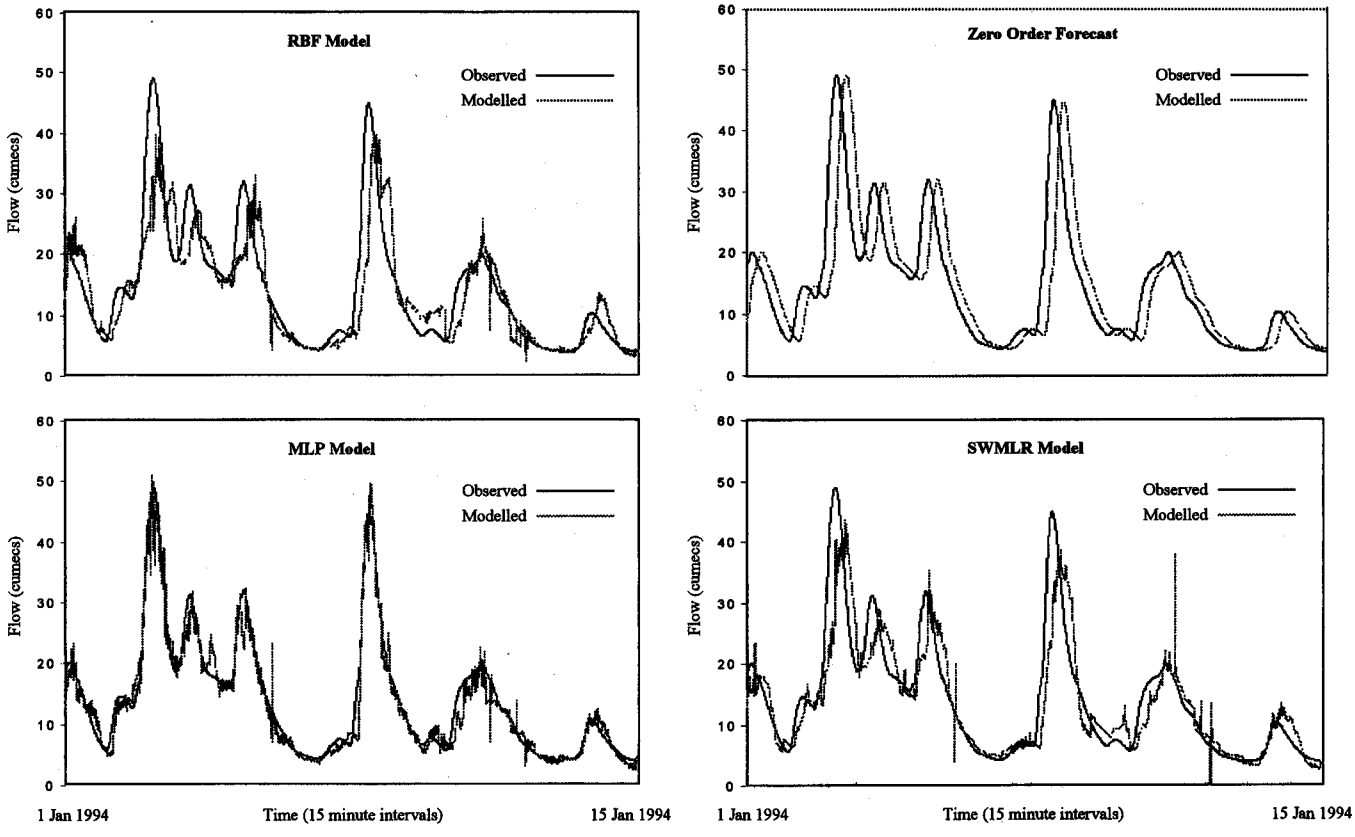

Fig. 3. *Validation results for first two weeks of 1994*

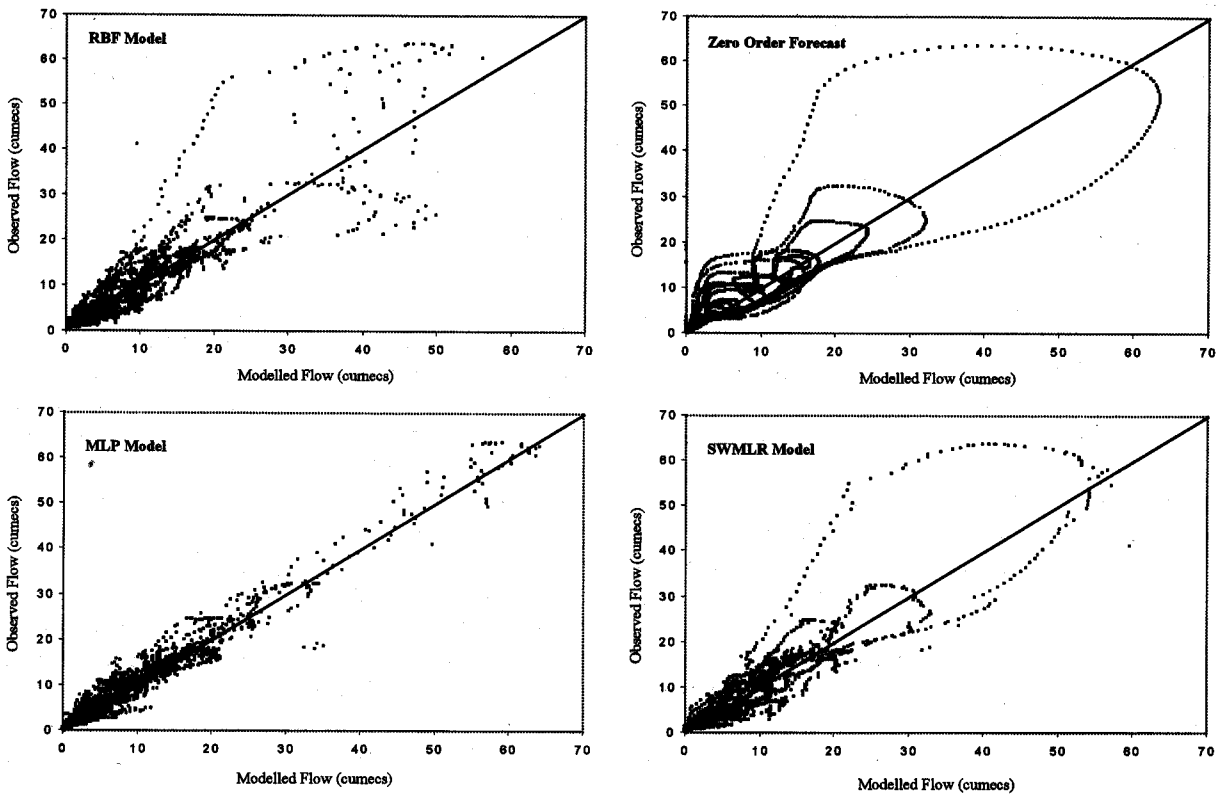

Fig. 4. *Validation results for autumn/early winter*

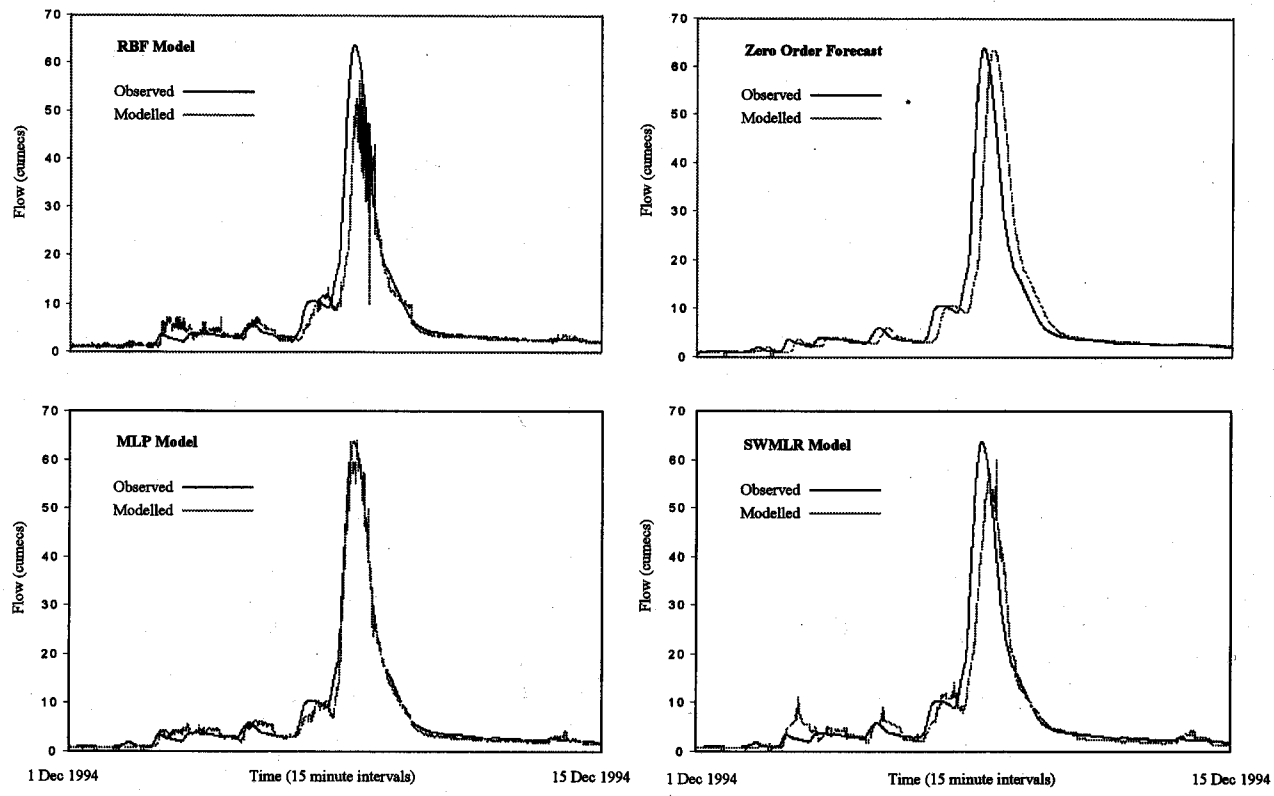

Fig. 5. *Validation results for penultimate two weeks of 1994*

confidence intervals were calculated for each model. Using all flows during the winter and early spring period, the standard error of the MLP model was 1.67 cumecs. Thus, when the MLP predicts a flow of 30 cumecs six hours hence, one is 95% confident (two standard errors) that flow will actually be 31 ± 3.34 cumecs (where 31 cumecs is calculated directly from the regression line based on the modelled prediction).

Unfortunately, due to heteroscedasticity these confidence intervals are valid only within certain ranges. This means that the variability of modelled versus observed flow cannot be guaranteed to be consistent over the entire range of flow ordinates. For example, in winter/early spring, for flows in excess of 20 cumecs the standard error increases to 3.4 cumecs. However, while this is greater than the overall standard error it is relatively small compared with the higher flow values. For example, Fig. 6 shows the ratio of the standard error to forecast flow (at 5 cumec intervals) for all models for both seasons. This figure shows how the standard error of each model decreases appreciably as flow levels increase and thus all models provide relatively accurate forecasts for flows in excess of 20 cumecs.

Discussion

Table 4 presents a summary of the most proficient ANN rainfall-runoff models reported by other authors using real hydrometric data (based on validation results). These results have also been selected because they present either r^2 or CE performance measures which are not biased by the scale of data used. However, the studies vary in choice of time step, lead times and catchment characteristics. These factors must also be considered when comparing studies of this type with perhaps greater emphasis being placed on the qualitative factors that are identified. Therefore, Table 4 also supplies information on the basin area, network type and structure (in terms of number of nodes in each layer), data standardisation approach and training algorithm used in each case.

The present results, together with those in Table 4, support the assertion that ANNs are well suited to rainfall-runoff modelling and flood forecasting. According to the r^2 statistic, the present MLP results compare favourably with all other studies listed in Table 4. Considering the coefficient of efficiency, CE, results by Minns and Hall (1997) and Campolo *et al.* (1999) are marginally better: 98 and 98.5% respectively, compared with 97% herein. However, these studies modelled flow with lead times of 30 minutes and one hour respectively as opposed to the 6 hour lead time in the present study and lead time is catchment specific and depends upon the time-scale of the phenomenon investigated. Campolo *et al.* (1999) suggest that the limiting time horizon corresponds to the minimum time lag between rainfall and rising water levels; this factor varies on an event by event basis in consequence of the inter-play between successive storm characteristics and the state of saturation of the catchment. In the case of the River Mole, the strongest (rainfall) predictor of discharge had a lag interval of 6 hours (see Table 1) implying that this was the mean response time of the catchment.

A comparison of the MLP results with those of Dawson and Wilby (1998) indicates that dramatic improvements in the validation statistics arise when ANNs are trained and tested against discharge data from the same season. The need for seasonal stratification of the training data implies that antecedent flows are not able to characterise fully the pre-storm conditions of the basin year-round. This is because other factors such as inter-seasonal variations in effluent discharges, vegetation and snow cover or soil moisture profiles may affect subsequent flood evolution.

A further consideration is the resources required to develop each neural network model. While the results presented in this paper indicate that the MLP is more accurate than other techniques, the length of time required for model training should not be discounted. For example, SWMLR and RBF models, built in a fraction of the time required for an equivalent MLP, can still provide acceptable predictions of flows in most cases (see Tables 2 and 3). This factor was also acknowledged by Fernando and Jayawardena (1998) who noted that RBFs require fewer parameters than the MLP (particularly when an orthogonal least squares algorithm is used to determine the number of basis functions). The RBF, therefore, requires less knowledge about the functioning of ANNs for their implementation than an MLP and might, therefore, be the first choice for ANN feasibility studies.

Conclusions

A comparison of the flow forecasting skill of two ANN configurations using a cross-validation technique and 15 minute rainfall-runoff data for the River Mole indicates that the multi layer perceptron model performed better than the radial basis function network. The RBF was, in turn, marginally more skillful than a less computationally demanding stepwise multiple linear regression model and zero order forecasts. The relatively poor results for the RBF were attributed to the clustering algorithm's selection of suboptimal cluster centres.

The preceding results suggest several avenues for further study. Firstly, as Table 4 indicates, direct comparisons between different ANN configurations and catchment studies are highly problematic. In this respect, the implementation of standard ANN descriptors and measures of accuracy—such as Bayes factors which penalise unnecessary model complexity (Kass and Raftery, 1995)—would greatly assist model inter-comparison. Equally, an index of catchment responsiveness (such as the mean lag interval between rainfall and runoff) would enable more objective comparisons of ANN forecasting

Table 4 Results of other artificial neural network rainfall runoff models tested against independent data.

Reference	Time Step	Lead Steps	River	Catchment Area (km²)	Best Result (%)	ANN Type	Structure (I-H1-H2-O)	Normalisation Range	Training Algorithm
Coefficient of efficiency (CE)									
Abrahart & Kneale (1997)	Hour	+1	Wye, Wales	11	96	MLP	3-16-14-1	[0,1]	BP
Abrahart & Kneale (1997)	Hour	+12	Wye, Wales	11	60	MLP	22-16-14-1	[0,1]	BP
Abrahart & Kneale (1997)	Hour	+6	Wye, Wales	11	56	MLP	22-16-14-1	[0,1]	BP
Braddock *et al.* (1998)	Day	+1	blind catchment	N/A	92	MLP	14-6-0-1	[-0.9,0.9]	P/C
Campolo *et al.* (1999)	Hour	+1	Tagliamento, Italy	2480	99	MLP	44-3-0-1	[0.1,0.9]	BP
Campolo *et al.* (1999)	Hour	+5	Tagliamento, Italy	2480	85	MLP	79-10-0-10	[0.1,0.9]	BP
Dawson & Wilby (1998)	15 Mins	+24	Amber, UK	139	87	MLP	15-10-0-1	[0,1]	BP
Dawson & Wilby (1998)	15 Mins	+24	Mole, UK	142	66	MLP	7-20-0-1	[0,1]	BP
Minns & Hall (1997)	30 Mins	+1	Silk Stream, UK	31.25	98	MLP	8-4-0-1	unknown	BP
Shamseldin (1997)	Day	0	Sunkosi, Nepal	18000	91	MLP	3-2-0-1	[0.1,0.85]	CG
Present Study (MLP)	15 Mins	+24	Mole, UK	142	93	MLP	7-20-0-1	[0.1,0.9]	BP
Present Study (RBF)	15 Mins	+24	Mole, UK	142	86	RBF	7-20-0-1	[0.1,0.9]	KMSVD
Coefficient of determination (r^2)									
Hsu *et al.* (1995)	Day	+1	Leaf River, USA	1949	95	MLP	9-3-0-1	[0.1,0.9]	LLSSIM
Poff *et al.* (1996)	Day	0	Independence, USA	230	84	MLP	2-?-0-1	unknown	BP
Poff *et al.* (1996)	Day	0	Little Patuxent, USA	97	83	MLP	2-?-0-1	unknown	BP
Thirumalaiah & Deo (1998)	Day	+2	Indravathi, India	41700	96	MLP	1-3-0-2	[0,1]	CG
Present Study (MLP)	15 Mins	+24	Mole, UK	142	96	MLP	7-20-0-1	[0.1,0.9]	BP
Present Study (RBF)	15 Mins	+24	Mole, UK	142	85	RBF	7-20-0-1	[0.1,0.9]	KMSVD

Key: BP (backpropogation); CG (conjugate gradient); I-H1-H2-O (input, first hidden, second hidden, output layer); KMSVD (K-Means clustering and singular value decomposition); LLSSIM (linear least squaes and multistart simplex optimisation); MLP (multi layer perceptron); P/C (combined pruning/constructive algorithm); (radical basis function).

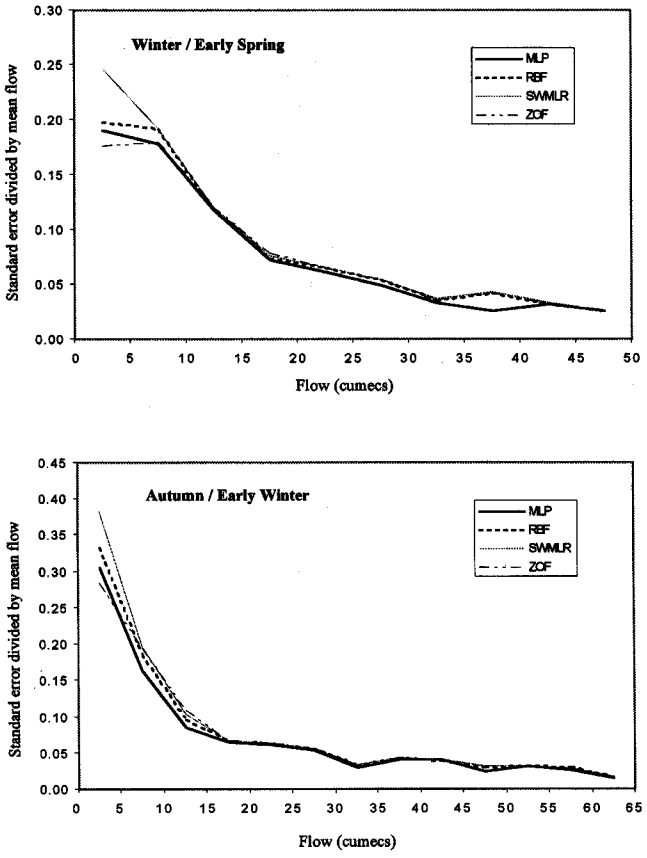

Fig. 6. *Standard error with respect to mean flow for all models*

skill at different lead times by recognizing that the limiting time horizon is catchment specific.

Secondly, the possibility of coupling ANN runoff models to runoff models should be explored as a means of extending forecast lead times, as in the Model Output Statistics (MOS) and Perfect Prog (PP) techniques (Klein and Glahn, 1974). For example, statistical downscaling methods could be used to transform coarse spatial resolution forecasts from numerical weather prediction models into local-scale input variables for hydrological modelling (Wilby and Dettinger, 1999; Hay *et al.*, 1999). In this case, the ANNs could even provide a means of handling inexplicit uncertainties introduced by the weather forecasts.

Thirdly, hybrid neural network models might be developed that maximise the advantages of different neural network architectures. For example, RBFs could be used to model intermediate flows and MLPs for flood forecasting. Furundzic (1998) found that a hybrid model of this type – which employed three MLPs to predict flow from data that were pre-clustered—performed better than a SWMLR model. In this approach, a self organizing map is used to precluster the data into different event types which are then passed through three separately trained neural network structures. The predictands are then recombined into a unified output.

Finally, as this study and others have shown, there is now a convincing basis for the pilot-testing of ANN rainfall-runoff models in real-time contexts. This of course presupposes the existence of telemetry and/or radar networks for generating the necessary ANN input drivers.

Acknowledgements

The authors are grateful to Richard Cross (Severn-Trent Environment Agency) and Brian Greenfield (Thames Environment Agency) for provision of the hydrometric data.

References

Abrahart, R.J. and Kneale, P.E., 1997. Exploring Neural Network Rainfall-Runoff Modelling, *6th National Hydrology Symposium*, Salford University, 9.35–9.44.

Battiti, R., 1992. First- and second-order methods for learning: between steepest descent and Newton's method, *Neural Computation*, **4**, 141–166.

Bishop, C.M., 1995. *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford, UK.

Braddock, R.D., Kremmer, M.L. and Sanzogni, L., 1998. Feed-forward artificial neural network model for forecasting rainfall run-off, *Environmetrics*, **9**, 419–432.

Campolo, M., Andreussi, P. and Soldati, A., 1999. River flood forecasting with a neural network model, *Wat. Resour. Res.*, **35**, 1191–1197.

Dai, H. and MacBeth, C., 1997. Effects of learning parameters on learning procedure and performance of a BPNN, *Neural Networks*, **10**, 1505–1521.

Dawson, C.W., 1996. A Neural Network Approach to Software Project Effort Estimation, *App. Artificial Intelligence in Eng.*, **1**, 229–237.

Dawson, C.W. and Wilby, R., 1998. An Artificial Neural Network Approach To Rainfall-Runoff Modelling, *Hydrol. Sci. J.*, **43**, 47–66.

Fernando, D.A.K. and Jayawardena, A.W., 1998. Runoff forecasting using RBF networks with OLS algorithm, *J. Hydrol. Eng.*, **3**, 203–209.

Furundzic, D., 1998. Application of neural networks for time series analysis: rainfall-runoff modeling, *Signal Processing*, **64**, 383–396.

Gallant, S.I., 1993. *Neural Network Learning and Expert Systems*, MIT Press, London

Hall, M.J. and Minns, A.W., 1993. Rainfall-Runoff Modelling as a Problem in Artificial Intelligence: Experience with a Neural Network, *BHS 4th National Hydrology Symposium*, Cardiff, 5.51–5.57.

Hay, L.E., Wilby, R.L. and Leavesley, G.H., 1999. Methods for predicting future climates and water budgets, including downscaling of GCMs. *Potential Consequences of Climate Variability and Change to the Water Resources of the United States*. American Water Resources Association, Atlanta, Georgia, USA, in press.

Hirose, Y., Yamashita, K. and Hijiya, S., 1991. Back-propogation algorithm which varies the number of hidden units, *Neural Networks*, **4**, 61–66.

Hsu, K., Gupta, H.V. and Sorooshian, S., 1995. Artificial Neural Network Modeling of the Rainfall-Runoff Process, *Wat. Resour. Res.*, **31**, 2517–2530.

Jayawardena, A.W., Achela, D. and Fernando, K., 1998. Use of Radial Basis Function Type Artificial Neural Networks for Runoff Simulation, *Computer-aided Civ. Infrastr. Eng.*, **13**, 91–99.

Karunanithi, N., Grenney, W.J., Whitley, D. and Bovee, K., 1994. Neural Networks for River Flow Prediction, *J. Comput. Civ. Eng.*, **8**, 201–220.

Kass, R.E. and Raftery, L.R., 1995. Bayes factors, *J. Am. Statist. Assoc.*, **90**, 773–795.

Klein, W.H. and Glahn, H.R., 1974. Forecasting local weather by means of model output statistics, *Bull. Am. Meteorol. Soc.*, **55**, 1217–1227.

Legates, D.R. and McCabe, G.J., 1999. Evaluating the use of "goodness-of-fit" measures in hydrologic and hydroclimatic model validation, *Wat. Resour. Res.*, **35**, 233–241.

Magoulas, G.D., Vrahatis, M.N. and Androulakis, G.S., 1997. Effective backpropagation training with variable stepsize, *Neural Networks*, **10**, 69–82.

Mason, J.C., Tem'me, A. and Price, R.K., 1996. A Neural Network Model of Rainfall-Runoff Using Radial Basis Functions, *J. Hydraul. Res.*, **34**, 537–548.

Mendenhall, W. and Sincich, T., 1995. *Statistics for Engineering and the Sciences* (4th edition), Prentice Hall International, London.

Minns, A.W. and Hall, M.J., 1997. Living with the Ultimate Black Box: More on Artificial Neural Networks, *6th National Hydrology Symposium*, Salford University, 9.45–9.49.

Refenes, A., Burgess, A.N. and Bents, Y., 1997. Neural networks in financial engineering: a study in methodology, *IEEE Trans. Neural Networks*, **8**, 1223–1267.

Rumelhart, D.E. and McClelland, J.L. (eds), 1986. Parallel Distributed Processing: Explorations in the Microstructures of Cognition', **1**, MIT Press, Cambridge.

Schalkoff, R.J., 1997. *Artificial Neural Networks*, McGraw Hill, New York.

Shamseldin, A.Y., 1997. Application of a neural network technique to rainfall-runoff modelling', *J. Hydrol.*, **199**, 272–294.

Song, X.M., 1996. *Radial Basis Function Networks for Empirical Modeling of Chemical Process*, MSc Thesis, University of Helsinki, http://www.cs.Helsinki.FI/~xianming (28 January 1999).

Thirumalaiah, K. and Deo, M.C., 1998. River stage forecasting using artificial neural networks, *J. Hydrol. Eng.*, **3**, 26–32.

Wilby, R.L. and Dettinger, M.D., 1999. Streamflow changes in the Sierra Nevada, California, simulated using a statically downscaled General Circulation Model scenario of climate change, In: McLaren, S. and Kniveton, D. (Eds.) *Linking Climate Change to Land Surface Change*, Kluwer Academic Publishers, Lancaster, UK, in press.

Zealand, C., M. Burn, D.H. and Simonovic, S.P., 1999. Short term streamflow forecasting using artificial neural networks, *J. Hydrol.*, **214**, 32–48.